TIERED EXPERT EMBEDDING: INPUT-SIDE EXPERT EMBEDDINGS FOR TIERED COMPRESSION OF MIXTURE-OF-EXPERTS LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) architectures scale model capacity without scaling per-token compute, but their massive expert parameters dominate memory and storage cost. Merging each layer's experts is the direct way to shrink them: several original experts are merged into one and share its intermediate dimension. This remains viable because of a general property of gated FFNs: for any given input, only about half of an expert's intermediate units pass the gating. The active subnetworks of merged members therefore rarely saturate the shared weights, and the residual merge error stays low-dimensional enough to be cancelled by a per-slot shift of the expert's *input*. We propose Tiered Expert Embedding (TEE), a simple compression framework that (i) merges each layer's experts with *tiered*, frequency-aware compression ratios, (ii) freezes the router and remaps original expert slots onto merged experts, and (iii) attaches a tiny per-slot *expert embedding* w_0 on the input side to compensate for merging error. On Qwen3.6-35B-A3B, TEE compresses expert parameters by $5.33\times$ and total parameters by $4.07\times$ while leaving per-token activated parameters unchanged. Even with *random* expert pairing at a uniform 2:1 ratio, the pipeline loses only 0.8 points on MMLU-Pro: at $2\times$ the capacity slack needs no pairing structure at all. At the full $5.33\times$ ratio the compressed model is comparable to the purpose-built Qwen3.5-9B dense model on MMLU-Pro (69.3 vs. 69.2) while activating $2.6\times$ fewer parameters per token, and outperforms MoLA under matched compression by $+8.7$ points.

1 INTRODUCTION

Mixture-of-Experts (MoE) layers scale language-model capacity without scaling per-token compute: a router sends each token to a few of many experts, so a model can hold hundreds of experts while each token activates only a handful. The bill arrives in memory. A modern fine-grained MoE LLM stores hundreds of experts per layer: our testbed, Qwen3.6-35B-A3B, stores 256 (34.66B parameters, 67GB in bf16) while each token activates only 3.45B, so the model does not fit on a single consumer GPU. This paper asks how much of that expert memory can be removed while leaving routing behavior, and hence per-token compute, intact.

Our starting point is a property of the experts themselves: in a gated expert, ReLU/SwiGLU-style gating zeroes out roughly half of the intermediate units for any given input, an activation sparsity well documented in trained transformers (Li et al., 2023; Luo et al., 2024) and already exploited inside MoE experts (Song et al., 2024). Each expert thus uses only a fraction of its capacity per input, so the active subnetworks of several experts could share one expert's weights, provided the sharers can still behave differently. What distinguishes them is the input: adding a small slot-specific vector to an expert's input shifts where each intermediate unit switches on, so the same weight matrix realizes a different active subnetwork for every slot. This input-side bias, one vector per original expert slot, is the mechanism that makes sharing possible.

How much sharing comes for free? We test the extreme case: pair experts uniformly at random, merge each pair's weights into one expert, attach the input-side embeddings, and refine them with light distillation. The resulting model loses only 0.8 points on MMLU-Pro relative to the teacher, with no clustering, no frequency statistics, and no pairing structure of any kind (Section 4.3). This

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

gating free lunch is free of structure (at 2:1, half-idle experts leave enough capacity slack that no pairing search is needed) but not free of training: zero-shot merging at the full $5.33\times$ ratio collapses (32.2 points; Table 2), and beyond $2\times$ the capacity slack runs out, so structure must take over.

Structure is what our method, Tiered Expert Embedding (TEE), adds beyond the free regime. Within each layer, experts are sorted by the routing mass they carry on a calibration corpus and merged at frequency-dependent ratios (2:1 for the heavily used head, up to 10:1 for the long tail, where we show redundancy concentrates), reducing 256 experts per layer to 48. Every original slot then keeps its own input-side embedding, refined together with the merged weights by on-the-fly distillation; the router is never retrained, so routing decisions and per-token compute are preserved by construction. The pipeline is a one-off, single-GPU procedure: one calibration pass over the MMLU-Pro development split, merging, then distillation training between 0.06% and 17.5% of parameters in under a GPU-day. It transfers across distributions at a cost of 1.6 points in our tests (Section 4.3).

TEE differs from prior MoE compression in where the compensation lives: low-rank adapter compensation (Gao et al., 2025; Wu et al., 2024) and prototype reassignment (Yao et al., 2026; Chakrabarti et al., 2026) both keep it on the output side, while our ablation identifies the input-side embedding as the component that carries the ratio beyond the free $2\times$ regime (removing it drops MMLU-Pro from 69.3 to 55.8). Weight-only quantization compresses bit-widths rather than removing experts and is complementary. Our contributions are threefold. **Observations:** expert activation mass is extremely skewed *per layer*, and redundancy grows along each layer's frequency ranking; a random-pairing experiment delineates the free $2\times$ regime. **Method:** TEE combines frequency-tiered merging, per-slot input-side embeddings, and two-phase distillation under a frozen router, preserving per-token compute by construction. **Findings:** at $5.33\times$ expert compression the student is comparable to a purpose-built 9B dense model on MMLU-Pro at $2.6\times$ fewer activated parameters per token; beyond $2\times$, embeddings and distillation, not pairing structure, carry the ratio.

On Qwen3.6-35B-A3B this yields an 8.51B-parameter student (67GB $\rightarrow$ 17GB in bf16) that runs on a single 24GB GPU.

2 Related Work

MoE compression. Expert pruning drops under-used experts from router statistics (Lu et al., 2024; Liu et al., 2024; Xie et al., 2024; Lasby et al., 2025; Bai et al., 2025); merging consolidates them via routing-aware clustering (Li et al., 2024; Chakrabarti et al., 2026), subspace projection (Li et al., 2026; 2025b), output merging (Miao et al., 2025), or delta decompression (Gu et al., 2025) (He et al., 2025; Muqeeth et al., 2024), sometimes paired with low-rank approximation (Huang et al., 2025; Yang et al., 2024) or applied at finer granularity (Xu et al., 2025; He et al., 2026). Which family wins is contested (Lasby et al., 2025; Jha et al., 2026), and pairwise merge scores can miss jointly mergeable triples (Zhong et al., 2026). These methods either drop experts outright, losing the routing attached to them, or merge at uniform ratios; TEE instead uses frequency-tiered per-layer ratios and preserves every original slot through a frozen router plus input-side embeddings, keeping the fine-grained routing control that merging otherwise gives up (Lasby et al., 2025; Yao et al., 2026).

LoRA-based expert approximation. MoE-LoRA PEFT methods (Gao et al., 2025; Wu et al., 2024; Li et al., 2025a; Dou et al., 2024; Zadouri et al., 2024) place compensation in low-rank adapters, whose per-expert outputs must be materialized at high bandwidth cost (measured in Section 5.5). TEE keeps full-rank merged experts and shifts compensation to the input side, avoiding adapter-style memory traffic entirely.

Parameter-efficient adaptation. Our expert embedding w_0 is an expert-conditional, input-side adapter, related in spirit to delta tuning (Ding et al., 2022), $(IA)^3$ (Liu et al., 2022), bias-only tuning (Ben-Zaken et al., 2021), representation shifts (Fu et al., 2022), and prompt/prefix tuning (Li & Liang, 2021; Lester et al., 2021), but tied to a static expert-remapping structure and used for *compression* rather than task adaptation; unlike LoRA (Hu et al., 2022) it adds no low-rank factorization and no extra matmul.

MoE architectures and routing. TEE targets modern fine-grained MoE LLMs (Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024; Qwen Team, 2026) whose load balancing (Wang et al., 2024a) flattens globally aggregated usage, a picture we show is misleading for compression (Section 5.1). Analyses of trained MoEs report token-determined routing that stabilizes early (Xue et al., 2024), neuron-level rather than domain-level specialization (Lo et al., 2024), and reusable expert collaboration groups (Tang et al., 2026), favouring merging by behaviour under a frozen router over re-tuning it.

Long-chain reasoning and budget forcing. Our calibration protocol borrows budget forcing from test-time scaling (Muennighoff et al., 2025) (Section 3.3).

3 METHOD

Figure 1 sketches the three stages developed below: calibration, tiered merging, and input-side embeddings with on-the-fly distillation.

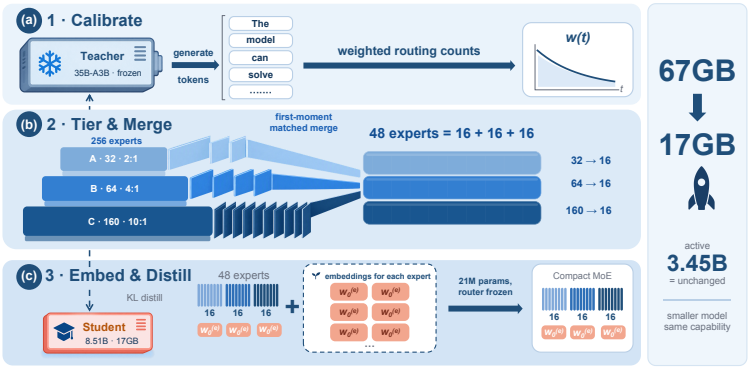

Figure 1: The TEE pipeline: per-layer expert-mass calibration, frequency-tiered merging under a frozen router, and per-slot input-side embeddings w_0 refined by on-the-fly distillation.

3.1 SETUP AND NOTATION

We consider an MoE language model with L MoE layers. Each layer contains E experts $\{F_e\}_{e=1}^{E}$ ($E = 256$) operating on hidden states $x \in \mathbb{R}^d$ ($d = 2048$); each expert is a SwiGLU FFN with intermediate size $d_{\text{ff}} = 512$. A router produces gating probabilities $p_e(x)$, and the layer output mixes the top-k experts ($k = 8$): $y = \sum_{e \in \mathcal{T}(x)} p_e(x)\, F_e(x)$, where $\mathcal{T}(x)$ is the top-k set. All non-expert weights and the router itself remain frozen throughout. Our goal is to compress each layer to $E' = 48$ physical experts (a $5.3\times$ reduction) *without* fine-tuning the router, so routing decisions, and hence per-token activated parameters, are preserved by construction. $E' = 48$ is the largest expert count for which the compressed bf16 checkpoint plus the KV cache of a 16k-token generation fits a single 24GB GPU.

3.2 MOTIVATING OBSERVATIONS

Three properties of the teacher's routing behavior, measured on calibration statistics (Section 3.3), dictate every design choice in TEE.

O1: Activation mass is extremely skewed per layer. Within each layer, the ratio between the top- and bottom-decile experts by weighted activation mass ranges from $4.1\times$ to $516.7\times$ across layers (median $44.8\times$; Section 5.1). A single uniform ratio cannot be right everywhere: it over-compresses the specialized head or wastes parameters on the redundant tail.

O2: Redundancy grows along the frequency ranking. Redundancy is not uniformly distributed. The mean pairwise cosine similarity of expert fingerprints (Section 3.4) climbs steadily along the

per-layer frequency ranking (Figure 2(e)), reaching 0.0095 among the rarest experts, about $2\times$ the within-layer null baseline (0.0046) and $1.5\times$ the head. Rarely used experts are behaviorally interchangeable; head and mid-frequency experts are functionally distinct, so the local compression ratio should grow with the local redundancy.

O3: Merging error is compensable on the input side. If g experts are merged into one physical expert F_j and every member keeps the same input, the members become indistinguishable. A mass-weighted average of the member weights cancels the first-order merge error on average over the calibration distribution (Section 3.4), leaving a per-token residual in the members' pre-activation differences. Since gating activates only about half of the intermediate units per input (Li et al., 2023), this residual is confined to the group's low-dimensional co-active subspace, where a single additive direction per member has room to act: a slot-specific shift of each member's *input*: one addition, no matrix multiplication, negligible parameter budget.

The three components of TEE map one-to-one onto these observations (frequency-tiered ratios for O1+O2, mass-weighted merging for O2, per-slot input-side embeddings for O3).

3.3 Activation Statistics with Terminal-Anchored Weighting

Merging decisions require reliable per-layer estimates of how much each expert matters. We run end-to-end generation on the MMLU-Pro validation split and hook every layer's router, recording each token's top-k expert indices and routing weights; collection is decoupled from weighting, so weighting schemes can be recomputed offline.

Budget forcing. Following test-time scaling practice (Muennighoff et al., 2025), when the CoT exceeds budget $T = 16384$ without terminating, we force a `</think>` token and let the model produce an answer. Forced samples are *kept and flagged* rather than discarded: the "salvage experts" they recruit must be represented in the merge (analyzed in Section 5.6). The budget matches the evaluation-time generation budget, so the calibration distribution, including its failure modes, mirrors deployment conditions.

Terminal-anchored weighting. Because pre-termination reasoning segments carry the highest information density, tokens are weighted by

$$w(t) = \begin{cases} \exp(-(t_{\text{end}} - t)/\tau), & t < t_{\text{end}} \text{ (CoT)}, \\ 1, & t \geq t_{\text{end}} \text{ (answer)}, \end{cases} \qquad \tau = 96, \tag{1}$$

where t_{end} is the position at which reasoning ends: as generation approaches it, per-layer routing stabilizes onto the answer-producing experts, so merging statistics should be dominated by this regime rather than early exploratory tokens. Retention varies by less than half a point for any $\tau \in [48, 192]$, and the exponential form discounts by soft reasoning steps rather than raw token distance. The weighted activation mass of expert e in layer l is then $m_e^{(l)} = \sum_{x_t} w(t)\ p_e^{(l)}(x_t)\ \mathbb{1}\big[e \in \mathcal{T}^{(l)}(x_t)\big]$. All statistics in TEE are computed *per layer*; Section 5.1 shows that globally aggregated frequencies are misleading.

3.4 Tiered Expert Merging

Within each layer, experts are sorted by $m_e^{(l)}$ and partitioned into three tiers (A/B/C hold 32/64/160 experts, merged at 2:1/4:1/10:1 into 16 experts each), following O1+O2: redundancy concentrates in rarely used experts, so the ratio grows with local redundancy (a per-layer, non-uniform budget of the kind explored for pruning (Bai et al., 2025)). Equal merged counts keep per-tier parameter allocation balanced and the merging schedule hardware-friendly.

Functional fingerprints. Experts are merged by *behavior*, not weight proximity: functional and weight similarity are largely decoupled in fine-grained MoE (Chakrabarti et al., 2026), and pairwise signals alone cannot certify that a whole group is jointly mergeable (Zhong et al., 2026). Each expert is represented by a fingerprint $\mu_e \in \mathbb{R}^d$, the mean of its outputs over reservoir-sampled calibration inputs (4096 per layer, at which the $1/\sqrt{N}$ estimation noise is negligible relative to inter-expert

distances). We cluster fingerprints with K-means under cosine distance and a hard size constraint: a Hungarian assignment enforces exactly the tier's target membership (2/4/10) per cluster.

First-moment matching. Merged weights are activation-mass-weighted averages of their members, applied to every expert matrix (gate, up, down): $\tilde{W} = \sum_{e \in \mathcal{C}} \alpha_e W_e$ with $\alpha_e = m_e / \sum_{e'} m_{e'}$, so the stacked perturbations $\Delta_e = W_e - \tilde{W}$ satisfy $\sum_e \alpha_e \Delta_e = 0$, and Taylor-expanding the member output $F_{W_e}(x)$ in Δ_e around $\tilde{W}$ (gating $\varphi \in C^2$) gives $\tilde{F}(x) = \sum_e \alpha_e F_{W_e}(x) + O(\sum_e \alpha_e \|\Delta_e\|^2)$. Note the delimiters of this statement: the cancellation holds under the calibration mass weighting α_e, not the token-dependent routing $p_e(x)$, and it concerns the average member function, not any single slot. The mass-weighted merge is thus the optimal *static* choice (minimal first-order error under the calibration distribution), while closing the per-token, per-slot gap is what the input-side embeddings learn (Section 3.5; measured in Table 2).

3.5 Expert Embeddings: Input-Side Compensation

Merging collapses several original expert slots onto one physical expert, destroying their behavioral differences. TEE compensates on the *input* side: each original slot e retains a bias vector $w_0^{(e)} \in \mathbb{R}^d$, and the layer output becomes

$$y = \sum_{e \in \mathcal{T}(x)} p_e(x)\, F_{j(e)}\big(x + w_0^{(e)}\big), \qquad j = \mathrm{remap}(e), \tag{2}$$

where $\mathcal{T}(x)$ and $p_e(x)$ are the frozen router's own top-k set and gating weights, and remap is a static 256→48 table built during merging. The sum runs over the same k=8 slots as in the teacher, so per-token FLOPs and activated parameters are unchanged. Colliding slots are served as rows of one batched multiplication with distinct shifts, the shared weights read from HBM once: collisions only *reduce* weight traffic, and no probability mass is ever merged across slots.

Random initialization. Zero initialization makes every member of a cluster compute the same function, a symmetry gradient descent cannot break (colliding slots receive identical gradients). We break the symmetry with a small random draw,

$$w_0^{(e)} \sim \mathcal{N}\big(0, \sigma^2 I_d\big), \qquad \sigma = 0.02, \tag{3}$$

giving $\|w_0^{(e)}\| \approx \sigma\sqrt{d} \approx 0.9 \approx 1\%$ of $\mathbb{E}\|x\|$: enough to separate the slots' gradients, small enough to stay on the teacher's operating point. The compensating directions are learned, not constructed (Phase A; Section 3.6).

3.6 On-the-Fly Distillation

Teacher and student share the frozen backbone in a single process: the original experts (`no_grad`) produce teacher logits, the merged experts (grad) produce student logits, and we optimize the full-vocabulary KL divergence with zero logits storage:

$$\mathcal{L} = \sum_{t \in \mathrm{CoT+answer}} w(t)\, \mathrm{KL}\big(\mathrm{softmax}\big(z_t^{\mathrm{teacher}}\big) \,\|\, \mathrm{softmax}\big(z_t^{\mathrm{student}}\big)\big)\,. \tag{4}$$

Three practical choices matter. Differential learning rates (10^{-3} for w_0, 10^{-5} for merged weights with 8-bit Adam) let the freshly initialized embeddings travel far while the first-moment-matched weights need only local refinement; two phases (w_0 first with merged weights frozen, then jointly) let the embeddings absorb coarse merging error before any weight moves. The on-policy corpus mixes the validation set with teacher CoT, and the loss is computed on CoT and answer tokens only, matching Section 3.3. Only 17.5% of parameters are trainable in the joint phase (0.06% in Phase A), so the pipeline runs on modest hardware.

4 Experiments

4.1 Setup

Base model. Qwen3.6-35B-A3B (Qwen Team, 2026): 40 MoE layers, 256 experts per layer, top-8 routing, hidden size 2048, 34.66B total / 3.45B activated parameters (vision tower untouched).

Table 1: Main results. All compressed variants share the frozen router and the default MMLU-Pro-val calibration (IFBench out-of-distribution).

Method	Experts/layer	Activated/token	MMLU-Pro	IFBench
Teacher (Qwen3.6-35B-A3B)	256	3.45B	74.6	63.7
Qwen3.5-9B (dense reference)	—	9B	69.2	67.3
Random 2:1 pairing (full pipeline)	128	3.45B	73.8	60.3
MoLA (5:1, r=4, all-linear)	51 + LoRA	3.45B	60.6	55.3
TEE (ours)	48	3.45B	**69.3**	58.7

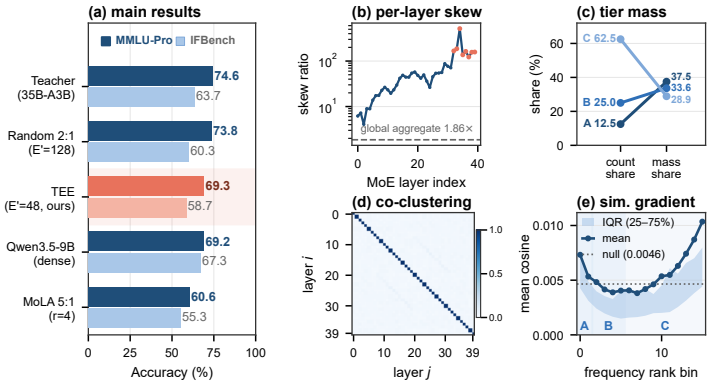

Figure 2: (a) Main results on MMLU-Pro and IFBench. (b) Per-layer expert load skew. (c) Expert-count versus routing-mass share by tier. (d) Cross-layer co-cluster retention. (e) Fingerprint cosine similarity along the per-layer frequency ranking.

Calibration and training. Activation statistics are collected on the MMLU-Pro validation split (the benchmark's designated development set, so calibration touches no test data; Appendix B) with budget forcing ($T = 16384$). Distillation trains on the same split with repeated sampling, mixed with teacher-generated CoT. This single student yields every number below unless noted (IFBench included, hence out-of-distribution); an IFEval-derived (Zhou et al., 2024) in-domain variant for IFBench is ablated in Section 4.3.

Evaluation. MMLU-Pro (Wang et al., 2024b) and IFBench (Pyatkin et al., 2025) (300 prompts over 58 held-out constraint types), without harness-assisted scaffolding: no quantization, maximal reasoning effort, and no code interpreter.[1]

4.2 Main Results

Table 1 and Figure 2(a) report the headline comparison. TEE compresses the teacher by $4.07\times$ in total parameters ($5.33\times$ in experts); the savings appear as memory (67GB $\rightarrow$ 17GB bf16, a single 24GB GPU) at unchanged per-token compute. The student loses 5.3 points to its teacher on MMLU-Pro but *is comparable to the purpose-built Qwen3.5-9B dense model* (69.3 vs. 69.2) at $2.6\times$ fewer activated parameters per token. On IFBench the 9B dense model leads both the student and the teacher (67.3 vs. 58.7/63.7), a gap inherited rather than introduced by compression, and out-of-distribution for the student ($+1.6$ in-domain; Section 4.3).

4.3 Ablation

Table 2 decomposes where the retained quality comes from. At 2:1, structure is free: the full pipeline with random pairing and a uniform 2:1 ratio loses only 0.8 points on MMLU-Pro, without clustering, tiering, or frequency statistics; with roughly half of each expert's units inactive per input (Li et al.,

[1]Our teacher numbers are lower than the official report, which uses an unpublished harness with a code interpreter; independent third-party measurements (artificialanalysis.ai) report 75 (teacher) and 69 (Qwen3.5-9B) on MMLU-Pro.

Table 2: Ablations. All variants use the default MMLU-Pro-val calibration except the last row (IFEval-derived, in-domain for IFBench).

Variant	Experts/layer	MMLU-Pro	IFBench
TEE (full)	48	69.3	58.7
random pairing, uniform 2:1 (no clustering/tiering)	128	73.8	60.3
merge + distill, w/o expert embeddings	48	55.8	42.6
zero-shot merge (mean, no training)	48	32.2	15.6
zero-shot merge (medoid, no training)	48	31.8	13.3
calibration source: IFEval-derived (in-domain)	48	—	60.3

2023), two active subnetworks fit into one expert's capacity, and at 2:1 the merge residual after first-moment matching is small enough for the embeddings to absorb without pairing structure (merging is not intrinsically free; the zero-shot rows show otherwise). Beyond $2\times$, every component becomes load-bearing: removing the expert embeddings drops the student to $55.8/42.6$ ($+13.5/+16.1$ over distillation alone; they make the $5.33\times$ ratio possible), and removing training collapses it to $32.2/15.6$. First-moment matching beats medoid weights ($+0.4/+2.3$), consistent with O3. Like every competitive MoE compressor (Li et al., 2025c; Kim et al., 2025) we train after structural compression, but place the residual on the input side, per slot, with a frozen router. Calibration source: the default student scores 58.7 on IFBench out-of-distribution; an IFEval-derived in-domain calibration adds $+1.6$ (60.3), still $+3.4$ above identically calibrated MoLA (55.3). MMLU-Pro calibration also keeps instruction following and math intact out-of-distribution (IFBench 58.7, GSM-8k 90.5; Appendix B); with a fixed application one simply calibrates on its data.

4.4 Comparison with LoRA-based MoE Approximation

We implement a matched LoRA-on-merged-experts baseline (all-linear, rank $r{=}4$, 5:1) in the adapter-traffic regime of MoE-LoRA methods (Gao et al., 2025; Wu et al., 2024), under the same calibration. TEE outperforms it by $+8.7$ MMLU-Pro and $+3.4$ IFBench (Table 1), widening to $+5.0$ with in-domain calibration (Section 4.3); Section 5.5 compares deployment cost. An expert-only merging baseline (Sub-MoE (Li et al., 2026)) at a comparable expert budget is in Appendix C.

4.5 Transfer to Nemotron-3.5-Lightning

The same recipe transfers to NVIDIA-Nemotron-3.5-Lightning (NVIDIA, 2026), a hybrid Mamba–MoE with 128 routed experts, one shared expert, top-6 routing, relu2 activations, and no gate projection. We compress its 23 backbone MoE layers to 24 routed experts, keeping the shared expert and the multi-token-prediction (MTP) MoE layer, yielding a 7.72B student ($4.09\times$ total, $5.33\times$ routed; 63GB $\rightarrow$ 15GB bf16). Table 3 reports 60.2 MMLU-Pro against the teacher's 65.0 (-4.8) and $+4.8$ over matched MoLA ($r{=}12$); IFBench follows the same order.

Table 3: Nemotron-3.5-Lightning. Shared expert retained; MTP MoE uncompressed.

Method	Experts/layer	Activated/token	MMLU-Pro	IFBench
Teacher	128+1	3B	65.0	55.7
MoLA (5:1, $r{=}12$)	25 + LoRA + 1	3B	55.4	50.7
TEE (ours)	24+1	3B	**60.2**	52.3

5 Analysis

The design of TEE rests on three observations about teacher routing (Section 3.2). This section revisits each one quantitatively: does the effect hold on the calibration data, and what breaks if the corresponding design choice is made differently?

5.1 Per-Layer Load Skew: Global Frequency Is an Illusion

Figure 2(b) plots, per layer, the ratio between the routing mass of the most- and least-loaded experts. Two views of the same data disagree sharply: aggregated over all layers, usage looks almost balanced (global ratio $1.86\times$, dashed line), while *within* a layer the ratio ranges from $4.1\times$ to $516.7\times$ (median $44.8\times$).

The gap is not a measurement artifact; it is the point. Skew is a property of each layer's routing distribution, not of global expert identity: the experts dominating layer l are not those dominating layer l', so cross-layer averaging mixes disjoint hot sets and cancels the skew. Statistics pooled over layers, including the global frequency rankings used by most merging and pruning baselines, describe a distribution no single layer follows. This is why TEE computes all statistics, tier boundaries, and merging decisions per layer (Section 3.3).

5.2 The Long Tail Is Not Free

If the tail of the frequency ranking carried negligible traffic, tail compression would be uninteresting: any ratio would do. Figure 2(c) splits expert *count* from routing *mass*: tier C is 62.5% of a layer's experts but still carries 28.9% of its routing mass: nearly one in three routed tokens touches a tail expert, so tail errors are not diluted away by the head.

This split explains both why the tail can be compressed hard and why it is not free. Tail experts are behaviorally the most redundant (O2), so many can share one physical expert with modest first-order error; but at 10:1, sixteen merged tail experts must jointly serve 28.9% of the layer's traffic, where the merge residual left after first-moment matching is no longer negligible: removing the per-slot embeddings that absorb it drops the student from 69.3 to 55.8 on MMLU-Pro (Table 2), and the zero-shot rows show the merge alone cannot serve the tail. The 10:1 tier is thus both the largest source of savings and the part that most needs input-side compensation (Section 3.5).

5.3 Merging Structure Is Strictly Layer-Specific

If merging structure were shared across depth, one could cluster once and amortize the decision over all layers. A necessary condition is that experts co-clustering in one layer also co-cluster in another. Figure 2(d) tests this: for every layer pair (i, j) it measures the fraction of expert pairs co-clustered in both. Off-diagonal retention sits at chance (0.026 versus 0.025 expected under independent clusterings), except for a weak band at layer distance $|i - j| \approx 12$ (band mean 0.037, at most 0.073; Appendix A).

Two conclusions follow. First, expert identity is layer-local: there is no cross-layer merging structure to exploit, so TEE's per-layer clustering and remapping is not merely safe but forced, and amortized or tied merging policies start from a structural deficit before any training begins. Second, because each layer is clustered independently, chance-level retention cannot be a clustering artifact: fine-grained expert specialization genuinely does not repeat across depth (the $\approx$12-layer band is too weak to change this but hints at a repeating functional motif; Appendix A).

5.4 Geometry of the Expert Embeddings

The embeddings w_0 are the only part of TEE learned from scratch, so it is worth verifying they start from nothing. Figure 4 (Appendix A) confirms the i.i.d. Gaussian signature at initialization: within- and between-cluster cosines coincide (centered at 0, spread $\approx 1/\sqrt{d} = 0.022$) and norms are uniform across depth ($\sigma\sqrt{d} \approx 0.91$). Hence initialization carries no cluster structure: slots that will collide onto one physical expert are geometrically indistinguishable from slots that will not, so any slot separation after training is learned by Phase A distillation, not inherited. The draw also validates the symmetry argument of Section 3.5 empirically: zero init would lock colliding slots to identical gradients, while $\sigma\sqrt{d} \approx 1\%$ of $\mathbb{E}\|x\|$ breaks the symmetry without leaving the teacher's operating point, so Phase A starts from an almost-exact first-moment match.

5.5 Deployment and Training Cost

TEE targets personal deployment (one user, one consumer GPU), where weight traffic comes from HBM (expert offloading (Cao et al., 2024; Liang et al., 2025; Zhou et al., 2025) is complementary). A merged expert is an ordinary dense FFN read once from HBM, while MoLA also fetches eight LoRA pairs per token; both are weight-bandwidth-bound at batch size 1, and TEE leads by $1.24\times$ at slightly lower peak memory (Table 4).

Table 4: Decode throughput and peak memory on one RTX 4090 (bf16, batch 1, median of three).

Method	tok/s	peak mem
MoLA (5:1, r=4)	23.5	18.1GB
TEE (ours)	**29.2**	**17.4GB**

Training is correspondingly cheap: the pipeline trains at most 6.06B parameters (17.5%; Phase A only 21M, 0.06%) on a frozen backbone ($\approx 10^{18}$ FLOPs, under a GPU-day), since the router and non-expert weights are frozen, first-moment matching lets Phase B start from weights already correct to first order, and the on-the-fly setup stores no teacher logits.

5.6 Budget Forcing and Salvage Experts

Forced-termination generations are *kept* in the calibration statistics rather than discarded as failures. The risk of keeping them is contamination; the risk of discarding them is worse: forced termination recruits "salvage experts" that rarely fire in normal reasoning (Section 3.3), so without forced samples these experts sink to the bottom of the mass ranking and are merged at the most aggressive ratio with the least representation, yet they carry disproportionate load exactly when the model approaches the generation budget, a regime that occurs at deployment because the calibration budget T matches the evaluation budget. Keeping the samples lets the merge see them at their deployment-time mass, while terminal-anchored weighting (Eq. 1) bounds their influence elsewhere; deployment cost is zero, so calibration mirrors deployment failure modes instead of flattering them.

6 Conclusion

Can a fine-grained MoE language model be compressed onto a single consumer GPU without retraining its router or raising per-token compute? We answer yes, provided three properties of teacher routing are respected: activation mass is extremely skewed *within* each layer while appearing balanced globally, so compression ratios must be tiered by per-layer frequency; redundancy concentrates in the tail of the ranking, but the tail still carries 28.9% of routing mass, so aggressive tail ratios must be paid for with explicit compensation; and the residual left by a first-moment-matched merge is low-dimensional enough to be absorbed on the input side by one learned embedding per original slot, with the router frozen. Tiered Expert Embedding (TEE) is built directly on these properties.

The properties prove sufficient, not just necessary. On Qwen3.6-35B-A3B, TEE cuts total and expert parameters by $4.07\times$ and $5.33\times$ (67GB to 17GB bf16, one 24GB GPU) at unchanged per-token compute; the student matches the purpose-built Qwen3.5-9B dense model on MMLU-Pro at $2.6\times$ fewer activated parameters, beats a matched LoRA baseline by $+8.7$ points, decodes $1.24\times$ faster, and transfers unchanged to Nemotron-3.5-Lightning at $4.09\times$. The ablations locate the retained quality where the analysis predicts: structure is free at 2:1; beyond that, every component is load-bearing.

Limitations and future work. Our claims are bounded in three directions. Quality depends on the calibration distribution: the default student transfers out-of-distribution with small but real losses, so fixed-domain applications should calibrate on their own data. The recipe presumes fine-grained experts with intra-expert redundancy: on Mixtral-8x22B, whose eight experts per layer provide neither, it loses 6.6 MMLU-Pro points at uniform 2:1 (Appendix D); the minimum granularity at which frequency tiering pays is open. Finally, the weak ≈12-layer co-clustering band (Appendix A) plays no role in TEE but, if amplified, may license the cross-layer merging policies that Section 5.3 rules out.

AI USE STATEMENT

We used generative AI tools for language polishing of this manuscript.

ETHICS STATEMENT

This work studies the compression of Mixture-of-Experts language models and does not involve human subjects, personal data, or deployment in safety-critical applications. We foresee no ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

The full compression pipeline is specified in Section 3, with all hyperparameters (calibration budget, tier ratios, and training schedule) listed in Sections 3.3–3.6 and 4.1. All per-layer statistics that drive the merging decisions are derived from the calibration generation described in Section 3.3. Code and the derived statistics needed to reproduce every figure will be released as supplementary material; the plotting scripts operate directly on the released data files. Appendix B documents the data hygiene of the development split, and Appendices A, C and D provide the additional analyses referenced from the main text.

REFERENCES

Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. Diep: Adaptive mixture-of-experts compression through differentiable expert pruning. *arXiv preprint arXiv:2509.16105*, 2025. doi: 10.48550/arXiv.2509.16105. URL https://arxiv.org/abs/2509.16105.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2021. doi: 10.48550/arXiv.2106.10199. URL https://arxiv.org/abs/2106.10199.

Shiyi Cao, Shu Liu, Tyler Griggs, Peter Schafhalter, Xiaoxuan Liu, Ying Sheng, Joseph E. Gonzalez, Matei Zaharia, and Ion Stoica. Moe-lightning: High-throughput moe inference on memory-constrained gpus. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024. doi: 10.1145/3669940.3707267. URL https://arxiv.org/abs/2411.11217.

Inesh Chakrabarti, Sourjya Roy, Bowen Bao, Thiago Crepaldi, Spandan Tiwari, and Ashish Sirasao. Shape mutating expert compression: Lorexperts and btexperts. *arXiv preprint arXiv:2608.07814*, 2026. doi: 10.48550/arXiv.2608.07814. URL https://arxiv.org/abs/2608.07814.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. doi: 10.48550/arXiv.2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14159*, 2021. doi: 10.48550/arxiv.2110.14168.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. doi: 10.48550/arXiv.2401.06066. URL https://arxiv.org/abs/2401.06066.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu,

Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022. doi: 10.48550/arXiv.2203.06904. URL https://arxiv.org/abs/2203.06904.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1932–1945, 2024. doi: 10.18653/v1/2024.acl-long.106.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. doi: 10.48550/arxiv.2101.03961.

Chin-Lun Fu, Zih-Ching Chen, Yun-Ru Lee, and Hung-yi Lee. Adapterbias: Parameter-efficient token-dependent representation shift for adapters in nlp tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022. doi: 10.48550/arXiv.2205.00305. URL https://arxiv.org/abs/2205.00305.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Ruibo Liu, Baochen Sun, Yawen Zhang, Daiyi Peng, Xiaoyuan Guo, and Vs Subrahmanian. MoLA: MoE LoRA with layer-wise expert allocation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5112–5127, 2025. doi: 10.18653/v1/2025.findings-naacl.284.

Hao Gu, Wei Li, Lujun Li, Qiyuan Zhu, Mark Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for MoE-based LLMs compression. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arxiv.2502.17298.

Haoze He, Xinkai Zou, Xuan Jiang, Xingyuan Ding, Ao Qu, Juncheng Billy Li, and Heather Miller. Less is moe: Trimming experts in domain-specialist language models. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. doi: 10.48550/arXiv.2606.05538. URL https://arxiv.org/abs/2606.05538.

Shwai He, Daize Dong, Liang Ding, and Ang Li. Towards efficient mixture of experts: A holistic study of compression techniques. *Transactions on Machine Learning Research (TMLR)*, 2025. doi: 10.48550/arXiv.2406.02500.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2106.09685.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts LLMs gains more. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.06270.

Saurav Jha, Maryam Hashemzadeh, Ali Saheb Pasand, Ali Parviz, Min-Joong Lee, and Boris Knyazev. Ream: Merging improves pruning of experts in llms. *arXiv preprint arXiv:2604.04356*, 2026. doi: 10.48550/arXiv.2604.04356. URL https://arxiv.org/abs/2604.04356.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. doi: 10.48550/arxiv.2401.04088.

Gyeongman Kim, Gyouk Chu, and Eunho Yang. Every expert matters: Towards effective knowledge distillation for mixture-of-experts language models. *arXiv preprint arXiv:2502.12947*, 2025. doi: 10.48550/arXiv.2502.12947. URL https://arxiv.org/abs/2502.12947.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. *arXiv preprint arXiv:2510.13999*, 2025. doi: 10.48550/arXiv.2510.13999. URL https://arxiv.org/abs/2510.13999.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. MixLoRA: Enhancing large language models fine-tuning with LoRA-based mixture of experts. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a. doi: 10.48550/arxiv.2404.15159.

Lujun Li, Qiyuan Zhu, Jiacheng Wang, Xiaoyu Qin, Wei Li, Hao Gu, Sirui Han, and Yike Guo. Sub-MoE: Efficient mixture-of-expert LLMs compression via subspace expert merging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22994–23002, 2026. doi: 10.1609/aaai.v40i27.39464.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arxiv.2310.01334.

Wei Li, Lujun Li, You-Liang Huang, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. MoE-SVD: Structured mixture-of-experts LLMs compression via singular value decomposition. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. doi: 10.18653/v1/2021.acl-long.353.

Zichong Li, Chen Liang, Zixuan Zhang, Ilgee Hong, Young Jin Kim, Weizhu Chen, and Tuo Zhao. Slimmoe: Structured compression of large moe models via expert slimming and distillation. *arXiv preprint arXiv:2506.18349*, 2025c. doi: 10.48550/arXiv.2506.18349. URL https://arxiv.org/abs/2506.18349.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, and Sanjiv Kumar. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2210.06313.

Jingcong Liang, Siyuan Wang, Miren Tian, Yitong Li, Duyu Tang, and Zhongyu Wei. Not all models suit expert offloading: On local routing consistency of mixture-of-expert models. *arXiv preprint arXiv:2505.16056*, 2025. doi: 10.48550/arXiv.2505.16056. URL https://arxiv.org/abs/2505.16056.

Enshu Liu, Junyi Zhu, Zinan Lin, Xuefei Ning, Matthew B. Blaschko, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. *arXiv preprint arXiv:2407.00945*, 2024. doi: 10.48550/arxiv.2407.00945.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 1950–1965, 2022. doi: 10.52202/068431-0142.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2024. doi: 10.48550/arXiv.2406.18219. URL https://arxiv.org/abs/2406.18219.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. doi: 10.18653/v1/2024.acl-long.334.

Yuqi Luo, Chenyang Song, Xu Han, Yingfa Chen, Chaojun Xiao, Xiaojun Meng, Liqun Deng, Jiansheng Wei, Zhiyuan Liu, and Maosong Sun. Sparsing law: Towards large language models with greater activation sparsity. *arXiv preprint arXiv:2411.02335*, 2024. doi: 10.48550/arXiv.2411.02335. URL https://arxiv.org/abs/2411.02335.

Ruijie Miao, Yilun Yao, Zihan Wang, Zhiming Wang, Bairen Yi, LingJun Liu, Yikai Zhao, and Tong Yang. Mergemoe: Efficient compression of moe models via expert output merging. *arXiv preprint arXiv:2510.14436*, 2025. doi: 10.48550/arXiv.2510.14436. URL https://arxiv.org/abs/2510.14436.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20275–20321, 2025. doi: 10.18653/v1/2025.emnlp-main.1025.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. Soft merging of experts with adaptive routing. *Transactions on Machine Learning Research (TMLR)*, 2024. doi: 10.48550/arXiv.2306.03745.

NVIDIA. NVIDIA-Nemotron-3.5-Lightning-30B-A3B, August 2026. URL https://huggingface.co/nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems 39: Datasets and Benchmarks Track*, 2025. doi: 10.52202/085713-1645.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Yixin Song, Haotong Xie, Zhengyan Zhang, Bo Wen, Li Ma, Zeyu Mi, and Haibo Chen. Turbo sparse: Achieving llm sota performance with minimal activated parameters. *arXiv preprint arXiv:2406.05955*, 2024. doi: 10.48550/arXiv.2406.05955. URL https://arxiv.org/abs/2406.05955.

Yuanbo Tang, Naifan Zhang, Yan Tang, Meixuan Chen, Shuhan Huang, Tingyu Cao, and Yang Li. Specialization through collaboration: Understanding expert interaction in mixture-of-expert large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026. doi: 10.18653/v1/2026.eacl-long.104.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024a. doi: 10.48550/arxiv.2408.15664.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuan Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Datasets and Benchmarks Track*, 2024b. doi: 10.52202/079017-3018.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of LoRA experts. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2404.13628.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*, 2024. doi: 10.48550/arXiv.2410.12013.

Yuzhuang Xu, Xu Han, Yuanchi Zhang, Yixuan Wang, Yijun Liu, Shiyu Ji, Qingfu Zhu, and Wanxiang Che. Camera: Multi-matrix joint compression for moe models via micro-expert redundancy analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. doi: 10.48550/arXiv.2508.02322. URL https://arxiv.org/abs/2508.02322.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024. doi: 10.48550/arXiv.2402.01739. URL https://arxiv.org/abs/2402.01739.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. MoE-I2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 10456–10466, 2024. doi: 10.18653/v1/2024.findings-emnlp.612.

Yilun Yao, Jiaming Pan, Elsie Dai, Peizhuang Cong, Yaoming Li, and Tong Yang. Conmoe: Expert-pool consolidation via prototype reassignment for moe compression. *arXiv preprint arXiv:2605.29350*, 2026. doi: 10.48550/arXiv.2605.29350. URL https://arxiv.org/abs/2605.29350.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient MoE for instruction tuning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arxiv.2309.05444.

Tao Zhong, Dongzhe Zheng, and Christine Allen-Blanchette. Hodgecover: Higher-order topological coverage drives compression of sparse mixture-of-experts. *arXiv preprint arXiv:2605.13997*, 2026. doi: 10.48550/arXiv.2605.13997. URL https://arxiv.org/abs/2605.13997.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2311.07911.

Yuxin Zhou, Zheng Li, Jun Zhang, Jue Wang, Yiping Wang, Zhongle Xie, Ke Chen, and Lidan Shou. Floe: On-the-fly moe inference on memory-constrained gpu. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arXiv.2505.05950. URL https://arxiv.org/abs/2505.05950.

A A Weak Periodicity in Cross-Layer Co-Clustering

Figure 2(d) reports cross-layer co-cluster retention on an absolute scale and shows that merging decisions cannot be shared across layers. Masking the diagonal and compressing the color scale (Figure 3) reveals one further, much weaker regularity that the absolute scale hides: layer pairs at distance $|i-j| \approx 12$ retain co-clustered expert pairs slightly more often than other pairs (band mean 0.037 vs. 0.026 for the remaining off-diagonal entries), and the strongest off-diagonal cells in the matrix (up to 0.073, about $3\times$ chance) concentrate on this band.

The effect is far too weak to exploit for compression as-is: even at its maximum, retention stays below 0.08, so the conclusion of Section 5.3, that expert merging is strictly layer-specific, is unaffected in absolute terms. We report it because a $\approx$12-layer periodicity in routing structure is unlikely to be a clustering artifact (the clustering is computed independently per layer), and it hints at a repeating functional motif across depth in this model family. Understanding its origin and whether the alignment can be amplified (e.g. by tying a small fraction of tier-C merges across the period) is a concrete direction for future work.

B Additional Benchmarks

We report two further benchmarks in Table 5. The student is the default MMLU-Pro-val-calibrated one of Table 1, so both benchmarks are out-of-distribution relative to calibration, the same transfer setting under which IFBench is reported there. A note on data hygiene: MMLU-Pro's 70-question validation split is the benchmark's designated development set (the official protocol reserves the 12,032-question test split for evaluation only), so using it for calibration and distillation is its intended use, not leakage; the test split is never touched outside evaluation.

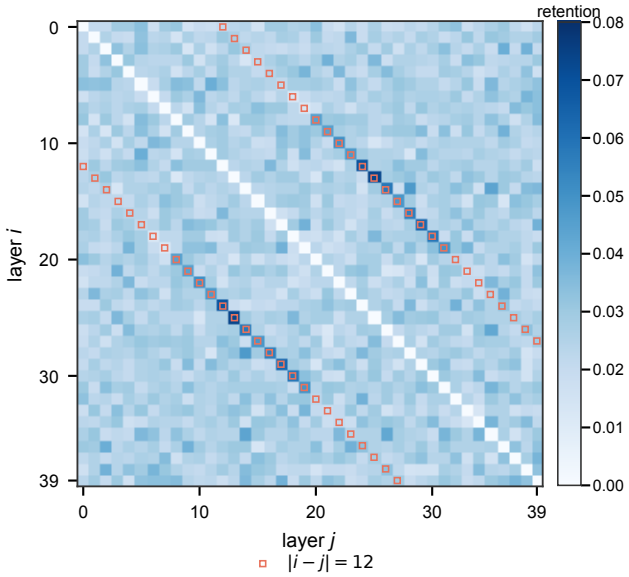

Figure 3: Cross-layer co-cluster retention with the diagonal masked and the color scale compressed to $[0, 0.08]$. Orange squares mark the two offset bands at $|i - j| = 12$; the darkest off-diagonal cells concentrate on these bands.

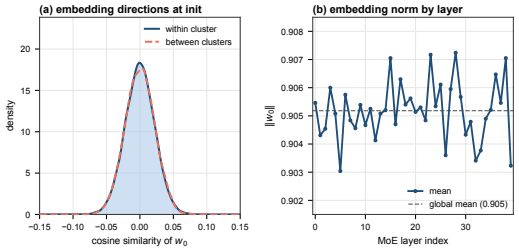

Figure 4: Expert embedding geometry at initialization (Section 5.4). (a) Within- versus between-cluster cosine similarity of w_0. (b) Embedding ℓ_2 norm across layers.

GSM-8k (Cobbe et al., 2021). The student retains 90.5 of the teacher's 91.4 (-0.9). We keep this benchmark in the appendix because it is effectively exhausted (frontier trillion-parameter models exceed 99.5), so a sub-point gap near the ceiling carries little signal in either direction.

HumanEval (Chen et al., 2021). pass@1 drops $82.3 \rightarrow 79.3$ (-3.0); pass@5 drops more, $86.6 \rightarrow 79.9$ (-6.7). A narrow pass@1/pass@5 gap is expected for CoT-style models (sampling adds little diversity once the reasoning pattern is set), but the larger pass@5 drop is an honest cost: the compressed model recovers less through test-time sampling than the teacher, i.e. part of what merging removes is exactly the diversity that pass@5 exploits. One further caveat: manual inspection of the generations found a non-zero number of solutions that pass the tests via example-specific special-casing while being incorrect in general; we report the scores as-is, for reference only.

C Expert-Only Reduction at Aggressive Ratios

Table 6 adds expert-only baselines at 64 experts per layer (removing $3/4$ of the teacher's experts, the most aggressive expert budget we evaluate): the merging representative Sub-MoE (Li et al., 2026) and the pruning representatives MoE-Pruner (Xie et al., 2024) and DiEP (Bai et al., 2025). The baselines are calibrated in-domain per benchmark, so their IFBench numbers enjoy an in-domain

Table 5: Supplementary benchmarks. Student calibrated on the MMLU-Pro validation split.

	GSM-8k	HumanEval	
	acc	pass@1	pass@5
Teacher (Qwen3.6-35B-A3B)	91.4	82.3	86.6
TEE (ours)	90.5	79.3	79.9

(IFEval-derived) calibration that our default student, calibrated on MMLU-Pro-val with IFBench out-of-distribution (Section 4.3), does not; the comparison is tilted in the baselines' favor, and TEE still leads every baseline on both benchmarks by at least $+8.0$.

Table 6: Expert-only reduction at an aggressive budget: merging (Sub-MoE) and pruning (MoE-Pruner, DiEP) at 64 experts per layer. Baselines are calibrated in-domain per benchmark; TEE numbers are reproduced from Table 1 (default MMLU-Pro-val calibration) for convenience.

Method	Experts/layer	MMLU-Pro	IFBench
Sub-MoE (Li et al., 2026)	64	54.3	34.3
MoE-Pruner (Xie et al., 2024)	64	25.3	36.0
DiEP (Bai et al., 2025)	64	45.7	50.7
TEE (ours)	48	**69.3**	**58.7**

Two caveats keep this comparison fair. First, none of the three baselines was validated beyond removing *half* of the experts in its original study; pushing them to $3/4$ removal on a 256-expert router places them far outside their validated ranges, so the comparison is only loosely fair and the gaps are not verdicts on any method at its intended ratios. Second, we cover both expert-only families (Sub-MoE for merging, MoE-Pruner and DiEP for pruning) because Sub-MoE already benchmarks against the earlier merge-then-compress and hard-core merging lines (MC-SMoE, HC-SMoE), so a single reproduction per family covers the behavior under aggressive budgets. The takeaway is narrow: at expert budgets where every expert slot must still be served, uncompensated expert-only reduction, merging or pruning, gives up 13–49 points to the teacher, and compensation (input-side, per slot, with a frozen router) is what makes 4–$5\times$ expert compression viable at all.

D Uniform 2:1 Compression on Mixtral-8x22B

Mixtral-8x22B-Instruct (Jiang et al., 2024) has eight experts per layer, too few for the 12.5/25/62.5 mass split of the Qwen testbed, so frequency tiers are undefined. We apply the full pipeline at a uniform 2:1 ratio (fingerprint clustering, quality-weighted merge, w_0, two-phase distillation), reducing eight experts to four at unchanged 39B activated parameters per token.

Table 7: Mixtral-8x22B-Instruct at uniform 2:1. Activated parameters are per token.

Method	Experts/layer	Activated/token	MMLU-Pro	IFBench
Teacher	8	39B	75.0	62.3
TEE (uniform 2:1)	4	39B	68.4	51.6

On Qwen the same 2:1 recipe with random pairing loses 0.8 MMLU-Pro (Table 1). On Mixtral it loses 6.6 on MMLU-Pro and 10.7 on IFBench. The gating free lunch of Section 4.3 depends on intra-expert redundancy at large expert counts; eight experts do not provide it.